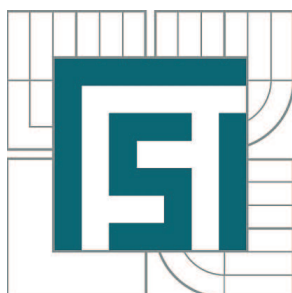


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA STROJNÍHO INŽENÝRSTVÍ**

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A  
BIOMECHANIKY**

FACULTY OF MECHANICAL ENGINEERING

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND  
BIOMECHANICS

## **ÚPRAVA DÁLKOVÉHO OVLÁDÁNÍ PRO CAR4**

CAR4 VEHICLE REMOTE CONTROL ENHANCEMENT

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**TOMÁŠ NEJEDLÍK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JOSEF VEJLUPEK**

BRNO 2012

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav mechaniky těles, mechatroniky a biomechaniky

Akademický rok: 2011/2012

## **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

student(ka): Tomáš Nejedlík

který/která studuje v **bakalářském studijním programu**

obor: **Mechatronika (3906R001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

### **Úprava dálkového ovládání pro Car4**

v anglickém jazyce:

#### **Car4 vehicle remote control enhancement**

Stručná charakteristika problematiky úkolu:

Úprava dálkového ovládní mobilního robotu Car4: doplnění displeje pro zobrazování hodnot. Dále drobné mechanické a elektrické úpravy na robotu - dle schopností studenta a domluvy s vedoucím práce.

Zadání je vhodné pro motivované a schopné studenty, kteří mají vůli se učit nad běžný rámec výuky.

Cíle bakalářské práce:

- 1/ shrnutí a výběr zobrazitelných údajů - výběr vhodného displeje.
- 2/ případná úprava elektroniky pro připojení displeje.
- 3/ naprogramování stávajícího dálkového ovládání (předpokládá se využití Kerhuel Toolboxu a automatického generování kódu z MATLABu):
  - 3.1/ přidání obousměrné komunikace
  - 3.2/ zobrazení dat na displeji
- 4/ úprava palubní elektroniky / kabeláže pro dobíjení.

Seznam odborné literatury:

VEJLUPEK, J. Konstrukce, simulační modelování a testování mobilního robotu s hybridním podvozkem. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009.

LAMBERSKÝ, V. Vývoj algoritm

Vedoucí bakalářské práce: Ing. Josef Vejlupek

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2011/2012.

V Brně, dne 16.11.2011

L.S.

---

prof. Ing. Jindřich Petruška, CSc.  
Ředitel ústavu

---

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.  
Děkan fakulty

# Abstrakt

Tato práce se zabývá připojením displeje k dálkovému ovládání a jeho programováním. Na začátku jsou popsány údaje, které je možné zobrazit na displej ze senzorů umístěných na experimentálním vozidle Car4. V práci je podrobně popsán přenos dat mezi vozidlem a dálkovým ovládáním, zpracování těchto dat a jejich zobrazení na displeji. Dále jsou popsány realizovatelné úpravy palubní elektroniky sloužící pro napájení vozidla Car4 a dobíjení akumulátorů.

# Abstract

---

This work deals with adding the LCD display to the remote control and its programming. At the introduction, we describe available data that we could get from the on-board sensors. Then we describe in detail how are the data transmitted between vehicle and its remote control, how the data are processed and then displayed. Also the modifications performed on the on-board electronic wiring and battery charging are discussed and described.

## Bibliografická citace

NEJEDLÍK, T. *Úprava dálkového ovládání pro Car4*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2012. 33 s. Vedoucí bakalářské práce Ing. Josef Vejlupek.

## Čestné prohlášení

---

Čestně prohlašuji, že jsem tuto práci vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím uvedené literatury.

Tomáš Nejdlík, Brno 2012

## Poděkování

---

Zde bych rád poděkoval všem, kteří přispěli ke vzniku této práce, kolegům z Mechlabu za rady a příjemné pracovní prostředí. Především děkuji svému vedoucímu bakalářské práce Ing. Josefu Vejlupekovi za jeho cenné rady, věnovaný čas a vedení bakalářské práce.

# Obsah

---

|       |   |    |
|-------|---|----|
| 1     | Úvod.....                                   | 9  |
| 2     | Displej.....                                | 10 |
| 2.1   | Základní rozdělení.....                     | 10 |
| 2.1.1 | Technologie LED.....                        | 10 |
| 2.1.2 | Technologie LCD.....                        | 10 |
| 2.2   | Připojení a programování displeje.....      | 11 |
| 2.3   | Zobrazitelné údaje.....                     | 14 |
| 2.3.1 | Enkodér.....                                | 14 |
| 2.3.2 | Proudový snímač LTS 25-NP.....              | 15 |
| 2.3.3 | ADIS 16045.....                             | 15 |
| 2.3.4 | Doplňková sensorika.....                    | 16 |
| 2.3.5 | Z ovládání.....                             | 17 |
| 2.3.6 | Stav baterie.....                           | 17 |
| 2.3.7 | Shrnutí zobrazitelných údajů.....           | 17 |
| 3     | Komunikace ovládání s vozidlem Car4.....    | 18 |
| 3.1   | Popis komunikace.....                       | 18 |
| 3.2   | Paketace.....                               | 18 |
| 4     | Úprava programu dálkového ovládání.....     | 20 |
| 4.1   | Současný stav – jednosměrná komunikace..... | 20 |
| 4.2   | Program dálkového ovládání.....             | 21 |
| 4.2.1 | Použité konfigurační bloky.....             | 22 |
| 4.2.2 | Problém práce displeje s UARTem.....        | 22 |
| 4.2.3 | Zpracování přijatých hodnot.....            | 23 |
| 4.2.4 | Vypisování hodnot na displej.....           | 26 |
| 5     | Úprava palubní elektroniky.....             | 28 |
| 5.1   | Akumulátory pro napájení.....               | 28 |
| 5.2   | Původní napájení.....                       | 28 |
| 5.3   | Napájení pomocí spínačů.....                | 28 |



# 1 Úvod

---

Pro zobrazování textových i obrazových informací se používají displeje, které předávají uživateli informaci z některých zařízení. V dnešní době se s displeji setkáme všude kolem nás, například při koupit jízdenky jsme pomocí displeje informováni o počtu vložených peněz a vybrané jízdenky. Na zastávkách jsme zase informováni o odjezdech dopravních prostředků. Displeje jsou použité také v řadě dalších aplikací, jako jsou mobilní telefony, měřicí zařízení, reklamy a další.

Cílem této bakalářské práce je doplnění dálkového ovládání o displej, na němž se budou zobrazovat informace pro uživatele experimentálního vozidla Car4. Zobrazena může být např. aktuální rychlost, ujetá vzdálenost a jiné hodnoty, které se musí z vozidla odeslat do ovládání. Možné zobrazitelné údaje na displeji budou jednotlivě popsány. Pro odesílání dat mezi dálkovým ovládáním a vozidlem Car4 bude přidána obousměrná komunikace. Pro programování dálkového ovládání a displeje se předpokládá využití automatického generování kódu z Matlabu. Model je vytvářen v prostřední Simulink, do kterého je nainstalovaný Kerhuel Toolbox obsahující potřebné bloky pro propojení mikrokontroleru a Simulinku.

Dalším cílem bakalářské práce je úprava palubní elektroniky, která slouží pro napájení vozidla Car4. Po úpravě se předpokládá snadnější spínání napájení a přepínání do stavu nabíjení akumulátorů umístěných na vozidle Car4.

## 2 Displej

---

### 2.1 Základní rozdělení

V dnešní době je pro zobrazování nejpoužívanější LED a LCD technologie.

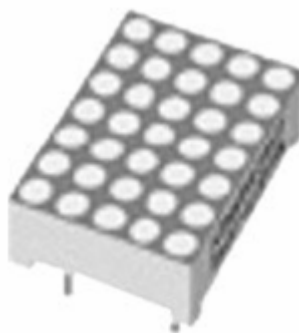
#### 2.1.1 Technologie LED

K zobrazování se používají LED diody, které jsou uspořádány do segmentů (většinou 7 nebo 16). Nejčastěji se používá sedmisegmentový displej pro zobrazování číslic. Při použití šestnáctisegmentového displeje se zobrazují i písmena. Číslice nebo písmena se zobrazují poskládáním jednotlivých segmentů, které jsou ovládány samostatně.

LED diody mohou být uspořádané také do matice. U maticových displejů je větší možnost zobrazených znaků, z jednotlivých bodů lze vytvořit číslice, písmena a další znaky.



Obr 2.1: Segmentový LED displej [5]

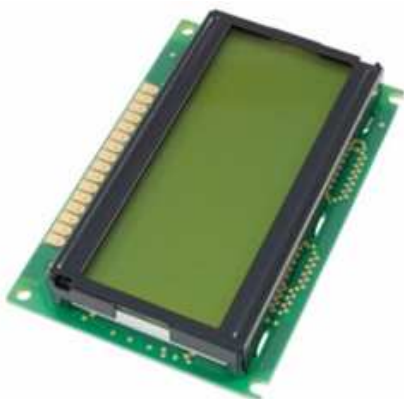


Obr 2.2: Maticový LED displej [5]

#### 2.1.2 Technologie LCD

LCD (Liquid Crystal Display) displej využívá k zobrazení tekuté krystaly, které elektrickým polem nebo vlivem tepla ovlivňují průchod světla. LCD displej je pasivní zobrazovací jednotka, potřebuje tedy zdroj světla. Jako zdroj světla lze využít okolní světlo, nebo se displej podsvětluje.

LCD displeje se dělí na alfanumerické a grafické. Alfaumerické displeje zobrazují alfanumerické údaje – písmena, číslice a základní znaky z ACII tabulky. Grafické displeje zase umožňují zobrazovat mimo písmen a číslic i různé křivky, čáry, loga, obrázky a zvláštní znaky.



Obr 2.3: Alfaumerický displej [5]



Obr 2.4: Grafický displej [5]

## 2.2 Připojení a programování displeje

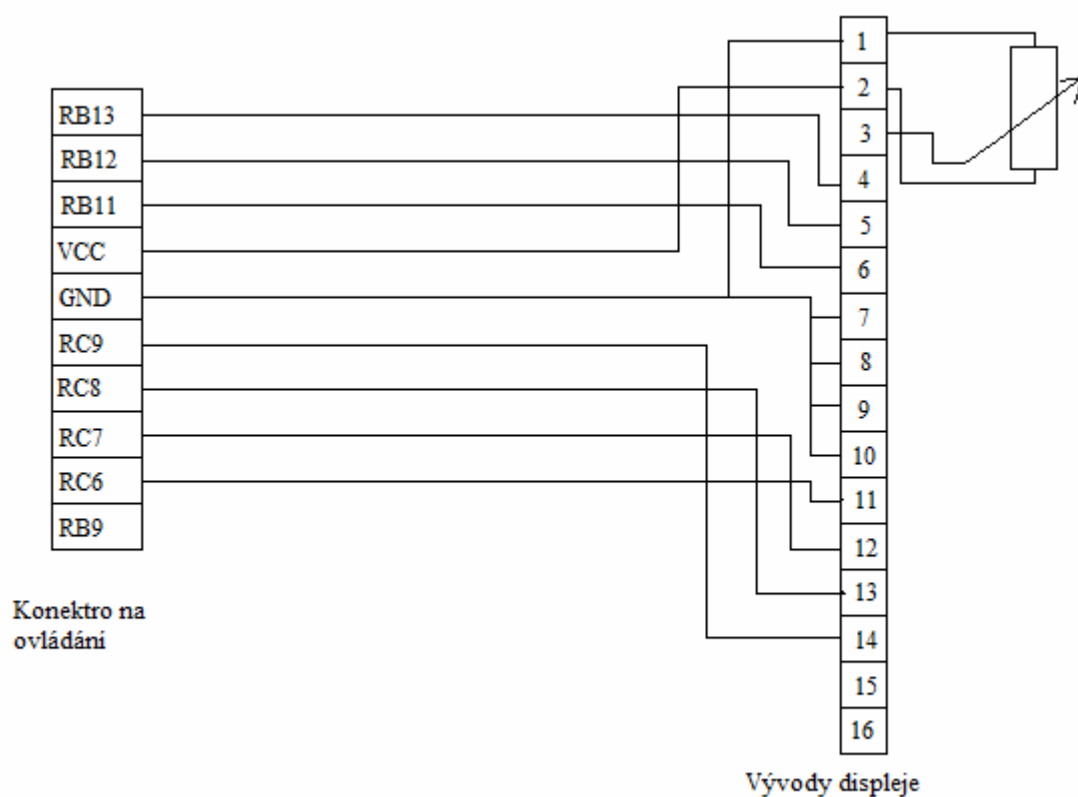
Pro připojení displeje k mikrokontroleru nebylo potřeba upravovat elektroniku již vytvořeného dálkového ovládání. Pro připojení displeje byl použit deseti pinový digitální výstup na dálkovém ovládání, který má dva piny napájecí (5V, GND) a zbylé slouží pro připojení displeje k mikrokontroleru. K připojení displeje se vytvořila pouze kabeláž, pomocí které se připojí displej k ovládání.

Pro připojení displeje k mikrokontroleru je potřeba čtyř nebo osmi datových vodičů, a to vodič s hodinovým signálem E, vodič pro přepínání zápisu dat nebo instrukcí RS, a pokud budeme z displeje i číst, tak ještě potřebujeme vodič R/W. Pro nastavení kontrastu displeje byl použit trimr, který je zapojen jako dělič napětí, jezdec je připojen na Vo.

Ovládání displejů je poměrně složité, proto se displeje dodávají s řadičem, který ovládá displej a komunikuje s okolím. Nejrozšířenější je řadič HD44780 od firmy HITACHI. Všichni výrobci používají řadič HD44780 nebo jeho ekvivalent, proto nezáleží od jakého výrobce displej je. Zapojení přívodního konektoru je stejné, což mnohem zjednodušuje práci s displeji. [3,6]

| Číslo vývodu | Označení | Význam                                  |
|--------------|----------|---|
| 1            | Vss      | 0V (napájení)                           |
| 2            | Vcc      | 5V (napájení)                           |
| 3            | Vee      | Nastavení kontrastu                     |
| 4            | RS       | 0-vstup je instrukce, 1-vstup jsou data |
| 5            | R/W      | 0-zápis do LCD, 1-čtení z LCD           |
| 6            | E        | Aktivace displeje                       |
| 7            | DB0      | Data, bit 1                             |
| 8            | DB1      | Data, bit 2                             |
| 9            | DB2      | Data, bit 3                             |
| 10           | DB3      | Data, bit 4                             |
| 11           | DB4      | Data, bit 5                             |
| 12           | DB5      | Data, bit 6                             |
| 13           | DB6      | Data, bit 7                             |
| 14           | DB7      | Data, bit 8                             |
| 15           |          | Napájení podsvětlení, anoda             |
| 16           |          | Napájení podsvětlení, katoda            |

Tab. 1: Význam vývodů displeje



Obr. 2.5: Připojení displeje k mikrokontroleru

Alfanumerické displeje se vyrábí v několika variantách 8x2, 16x1, 16x2, 16x4, 20x2, 20x4, 24x2, 24x4, 40x2, 40x4, které se liší počtem řádku a sloupců. Pro ukládání zobrazených znaků používá řadič HD44780 paměť DDRAM, která má velikost pro 80 znaků. Pro displeje, které mají více jak 80 znaků, se používají dva řadiče a mimo vývody E jsou všechny vývody spojeny paralelně, vývody E1 a E2 ovládají každý polovinu displeje. Každý znak je zobrazován jako matice 5x8 bodů (5x10 bodů) a uložen v paměti CGRAM.

| Příkaz                                       | RS | R/W | DB7  | DB6            | DB5            | DB4 | DB3 | DB2 | DB1 | DB0 | Popis  | Čas vykonání |
|--|----|-----|------|----------------|----------------|-----|-----|-----|-----|-----|--|--------------|
| Smaže displej                                | 0  | 0   | 0    | 0              | 0              | 0   | 0   | 0   | 0   | 1   | Smaže displej a nastaví kurzor na pozici 0.  | 1.64mS       |
| Nastaví kurzor na začátek                    | 0  | 0   | 0    | 0              | 0              | 0   | 0   | 0   | 1   | *   | Nastaví kurzor na pozici 0 a vynuluje posun displeje a   | 1.64mS       |
| Nastaví vstupní režim                        | 0  | 0   | 0    | 0              | 0              | 0   | 0   | 1   | I/D | S   | Určí směr pohybu kurzoru (I/D) a posun displeje (S). Tyto operace se provádějí během čtení/zápisu.       | 40uS         |
| Zapne/vypne displej, kurzor a jeho blikání   | 0  | 0   | 0    | 0              | 0              | 0   | 1   | D   | C   | B   | Zapíná/vypíná displej (D), kurzor (C) a jeho blikání (B).  | 40uS         |
| Nastaví pohyb kurzoru/displeje               | 0  | 0   | 0    | 0              | 0              | 1   | S/C | R/L | *   | *   | Nastaví pohyb kurzoru nebo displeje (S/C) a směr pohybu (R/L). Obsah DDRAM zůstane beze změny.           | 40uS         |
| Nastavení interface                          | 0  | 0   | 0    | 0              | 1              | DL  | N   | F   | *   | *   | Nastaví délku interface (DL), počet řádků displeje (N) a znakový font (F).                               | 40uS         |
| Nastaví pozici v CGRAM                       | 0  | 0   | 0    | 1              | Adresa v CGRAM |     |     |     |     |     | Po tomto příkazu jsou data ze vstupu zaznamenávána do CGRAM namísto DDRAM.                               | 40uS         |
| Nastaví pozici v DDRAM                       | 0  | 0   | 1    | Adresa v DDRAM |                |     |     |     |     |     | Po tomto příkazu jsou data ze vstupu zapisována do a čtena z DDRAM.                                      | 40uS         |
| Čte příznak BUSY a hodnotu adresového čítače | 0  | 1   | BF   | DDRAM addressa |                |     |     |     |     |     | Čte příznak BUSY (BF) indikující, že displej ještě provádí některou operaci, a pozici ukazatele adresy . | 0uS          |
| Zapíše do DDRAM nebo CGRAM.                  | 1  | 0   | data |                |                |     |     |     |     |     | Zapíše data ze vstupu DDRAM nebo do CGRAM.   | 40uS         |
| Čte data z DDRAM nebo z CGRAM.               | 1  | 1   | data |                |                |     |     |     |     |     | Čte data z aktuální adresy DDRAM nebo CGRAM.   | 40uS         |

Tab. 2: Příkazy řadiče displeje [6]

Po zapnutí mikroprocesoru a displeje se musí provést inicializace tohoto displeje, nastaví se 4 bitová komunikace (je používán 10-ti pinový konektor, proto byla zvolena 4 bitová komunikace), komunikace probíhá 2x pomaleji než 8 bitová. Nejprve se odešlou horní 4 bity a poté spodní 4 bity. Dále se při inicializaci nastavuje počet řádků, sloupců, směr psaní znaků, posun řádků, velikost písmen, pozice a blikání kurzoru. Jednotlivé instrukce, které displej přijímá, zabere nějaký čas, než je řadič zpracuje a do té doby nemohou být přijímány další instrukce. Časové intervaly, které potřebuje řadič pro zpracování jednotlivých instrukcí, jsou uvedeny v tabulce.

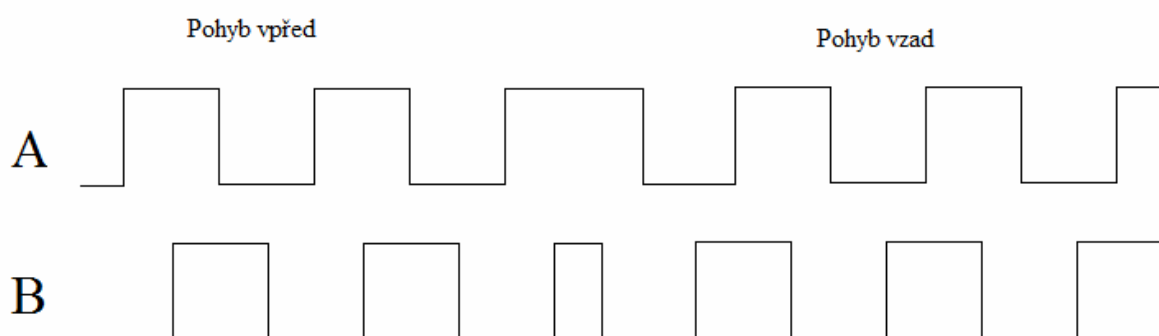
## 2.3. Zobrazitelné údaje

Tato kapitola je věnována popisu údajů z experimentálního vozidla Car4, které je možné zobrazit na displeji dálkového ovládání. Na displej mohou být zobrazeny hodnoty z enkodéru, z proudového čidla LEM, z čidla ADIS, které obsahuje magnetometr, gyroskop a akcelerometr. Dále údaje z kolizních čidel, teplota motoru, poloha tlumičů a hodnoty z dálkového ovládání.

### 2.3.1 Enkodér

Enkodéry jsou snímače, které slouží ke zjišťování polohy, rychlosti a zrychlení. Enkodér převádí rotační pohyb na digitální impulsy. Využívají se v aplikacích, kde je potřeba přesného řízení polohy, natočení, rychlosti, např. u robotu je potřeba pro jeho řízení znát polohu, rychlost. Enkodéry se užívají jako zpětná vazba pro řízení pohonů. Můžeme se s nimi setkat i při běžných činnostech, např. ovládání hlasitosti na Hifi soustavě, nebo u počítačové myši.

Enkodér obsahuje kolečko s obdélníkovými otvory, které jsou rovnoměrně rozmístěné a mají stejnou velikost. Otočení o jeden obdélník odpovídá o stejný počet stupňů. Přes obdélníkové otvory je vysíláno světlo ze světelného zdroje a za kolečkem je světlo přijímáno optickým snímačem. Tím získáme elektrické impulsy a obdélníkový signál. Z tohoto údaje zjistíme polohu nebo rychlost otáčení, ale neznáme směr. Proto se užívá ještě jeden snímač, který je posutý o půl vzdálenosti okénka. Tím tedy enkodér generuje dva obdélníkové signály, které jsou fázově posunuté. Znázorněno na obrázku č 2.6. [7]



Obr. 2.6: Průběh výstupního signálu z enkodéru

Výstup z periferie enkodéru je 16-bitové číslo udávající polohu odvozenou z počtu hran. Jde o relativní polohu motoru. Ta se pak přepočte přes počet pulsů enkodéru na otáčku, převodový poměr a poloměr kola na ujetou vzdálenost. Při použití derivace se získá rychlost a po další derivaci zrychlení. To lze získat i úhlových veličinách, jako je natočení, úhlová rychlost a úhlové zrychlení.

Pro naše účely jsme se rozhodli zobrazit na displej ujetou vzdálenost a rychlost.

### 2.3.2 Proudový snímač LTS 25-NP

Snímač slouží pro elektronické měření stejnosměrných, střídavých, pulsních a smíšených proudů. Senzor poskytuje analogový napěťový signál, který je úměrný proudu, který prochází motorem. Má galvanicky oddělené primární (vysoký výkon) a sekundární okruh (elektronický obvod). Ke snímání využívá princip Hallova jevu. [9]

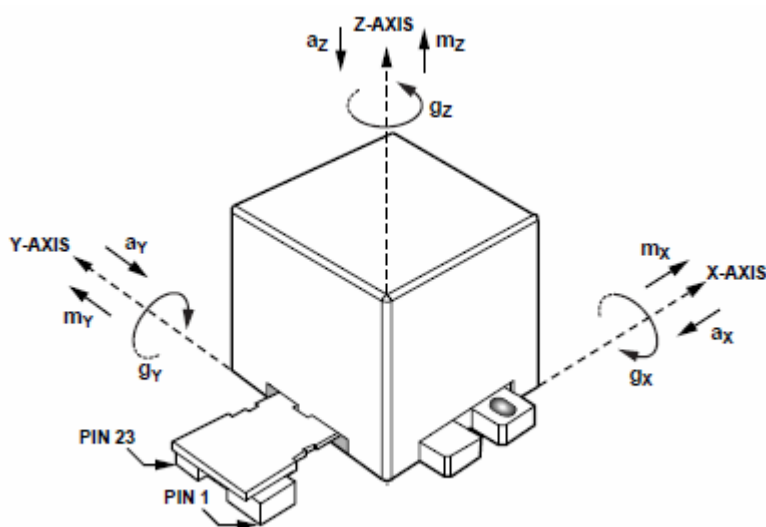
*„Hallův jev je proces generace Hallova elektrického pole v polovodiči za současného působení vnějšího elektrického i magnetického pole. Důsledkem toho se hromadí na jedné straně látky záporný náboj a na straně druhé náboj kladný. Díky tomu, že póly mají různý potenciál, vzniká Hallovo napětí.“ (citováno z[8])*

Získané napětí z proudového snímače je zpracováno mikrokontrolerem, pomocí AD převodníku je převedeno na digitální signál.

Z proudového čidla je možné zobrazit hodnotu proudu procházejícího motorem. A po vhodném přepočtu lze zobrazit moment.

### 2.3.3 ADIS 16045

Inerciální jednotka ADIS 16045 je součástka od firmy Analog Devices, která zahrnuje tříosý akcelerometr, tříosý gyroskop a tříosý magnetometr. Ke komunikaci s mikroprocesorem užívá sériové rozhraní SPI, pomocí kterého lze provádět nastavení a řízení senzoru. Senzor obsahuje teplotní čidlo a je kompenzován v rozsahu  $-40\text{ }^{\circ}\text{C}$  až  $+85\text{ }^{\circ}\text{C}$ . Pro spuštění měření není potřeba inicializace, ani žádný startovací příkaz, senzor zahájí měření po cca 220 ms po zapnutí napájení. Podle datasheetu je pro optimální výkon nastavena výchozí vzorkovací frekvence na 819,2 vzorku za sekundu. [10]



Obr.2.7: ADIS 16045 (převzato z [10])

Akcelerometr je senzor pro měření zrychlení. *Síla způsobující vibrace nebo změnu pohybu (akceleraci) působí na hmotu snímače, která pak stlačuje piezoelektrický prvek generující elektrický náboj úměrný stlačení. Protože je elektrický náboj úměrný síle a hmota snímače je konstantní, je tedy elektrický náboj také úměrný zrychlení - akceleraci. (citováno z [11])*

Na zemi je tíhové zrychlení  $g = 9,81 \text{ ms}^{-2}$ , které je kolmé k zemskému povrchu. Je-li akcelerometr v klidu a umístěn jednou jeho osou kolmo k zemi, naměří na této ose zrychlení 1 g. Senzor měří zrychlení v jednotkách g.

Gyroskop je senzor pro měření úhlové rychlosti. Využívá se v navigaci, např. u letadel. Integrací úhlové rychlosti získáme natočení.

Magnetometr je senzor pro měření směru a intenzity magnetického pole. Můžeme ho využít jako kompas. Využívá magnetorezistivního jevu, který funguje na principu změny odporu magnetického materiálu vlivem působení magnetického pole.

### 2.3.4 Doplnková sensorika

#### Teplota motoru

Na každém DC motoru u vozidla Car4 je připevněna destička s teplotním senzorem TMP275, který slouží pro měření teploty pláště DC motoru. Senzor komunikuje s mikrokontrolerem pomocí I2C sběrnice. Z naměřené hodnoty teploty pláště s pomocí odhadných konstant z matematického modelu je odhadnuta teplota vinutí DC motoru. Teplotní sensorika je u vozidla z bezpečnostních důvodů, pokud by DC motor zablokoval a neotáčel se, protékaly by jím velké proudy a motor by se začal přehřívat. [2]

#### Poloha tlumičů

Pro měření polohy tlumičů u vozidla Car4 jsou využity ohybové senzory, které mění při ohýbání svůj odpor a tím i napětí (snímač je zapojen do odporového děliče). Napětí je zpracováno AD převodníkem na digitální signál a posíláno I2C sběrnici do mikrokontroleru, kde je napětí převedeno na hodnotu stlačení tlumiče. [2]

#### Kolizní senzory

Vozidlo Car4 je vybavené kolizními senzory, které slouží k zabránění srážky vozidla s překážkou. Jsou použity IR snímače, které pracují na principu snímání odraženého záření od detekovaného předmětu. Signál je převeden AD převodníkem na digitální a sběrnici I2C veden do mikrokontroleru. [2]

Hodnoty ze senzorů je možné zobrazit na displej, ale v současné době vozidlo Car4 není vybaveno doplňkovými senzory, a proto nebudou tyto hodnoty využity pro zobrazení na displej.



### 2.3.5 Z ovládání

Z dálkového ovládání, které slouží pro bezdrátové ovládání vozidla Car4, lze zobrazit údaje o požadovaném řízení. Vozidlo Car4 je řízené velikostí momentu a natočení přední a zadní nápravy, tyto hodnoty je možné zobrazovat na displeji. Při řízení vozidla je možné zvolení modu řízení a řazení rychlostních stupňů, které umožňuje nastavení rychlostního rozsahu pro řízení. Na displej je tedy možné zobrazit zvolený rychlostní stupeň a nastavený mód řízení. Výhodou zobrazených údajů z ovládání je, že se hodnoty nemusejí odesílat z jednoho mikrokontroleru na druhý.

### 2.3.6 Stav baterie

Na displeji je možné zobrazit stav akumulátorů umístěných na vozidle Car4 a dálkovém ovládání. Jelikož napětí na akumulátorech má vyšší hodnotu, než s kterou pracuje mikrokontroler, použije se odporový dělič užívající se pro získání výstupního napětí, které je přímo úměrné vstupnímu napětí. Dále je napětí přes AD převodník převedeno na digitální signál, a ten je zpracován mikrokontrolerem.

### 2.3.7 Shrnutí zobrazitelných údajů

Ze senzorů umístěných na vozidle Car4 jsou na displeji zobrazeny z enkodéru údaje o ujeté vzdálenosti a rychlosti jízdy. Z proudového čidla Lem je zobrazen záběrný moment přední a zadní nápravy. Z inerciální jednotky ADIS je zobrazen údaj z magnetometru jakým směrem je vozidlo natočeno, neboli azimut. Pro zobrazení těchto hodnot byl zvolen alfanumerický displej 20x4 s dostatečným počtem znaků pro námi chtěné zobrazené údaje.

## 3. Komunikace ovládání s vozidlem Car4

### 3.1 Popis komunikace

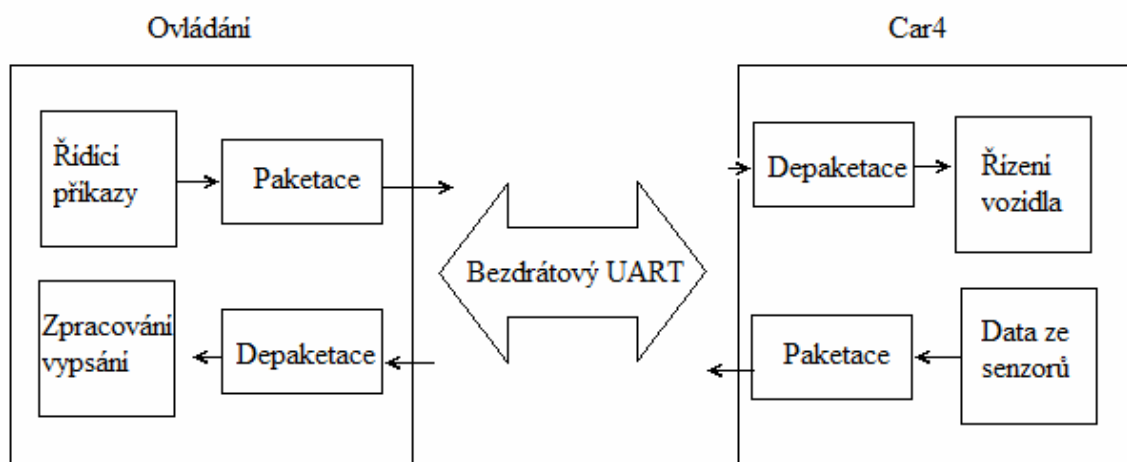
Komunikace mezi experimentálním vozidlem Car4 a dálkovým ovládáním je uskutečněna pomocí bezdrátového UARTu (Universal Asynchronous Receiver / Transmitter). Pro bezdrátový přenos je využit modul HW 86010, který umožňuje plně duplexní obousměrný datový přenos na vzdálenost až 7 km. Modul je vyroben firmou Höft & Wessel. Přenos je uskutečněn sériovou linkou, která je při připojení vhodných budičů kompatibilní s RS232, pomocí které je modul připojen k mikrokontroleru. Pro naše účely jsou využity dva moduly HW 86040. Jeden pro Car4 a druhý pro dálkové ovládání.

Linka RS232, pomocí které je modul připojen k mikrokontroleru, slouží pro sériovou komunikaci, bity přenášených dat jsou posílány sériově za sebou. Přenos může být simplexní (jednosměrná komunikace), plně duplexní (obousměrná komunikace) nebo poloduplexní, kdy jsou vysílací a přijímací pin spojeny a komunikace může probíhat jedním nebo druhým směrem, nemůže probíhat oběma směry současně. Vysílací a přijímací strany se musí pomocí nějakého protokolu domluvit, pokud chtějí vysílat druhým směrem.

Data jsou vysílány na Tx (Transmitter) pin a přijímány Rx (Receiver) pinem na mikrokontroleru.

Pro odesílání dat z ovládání do Car4 stačí spojit vysílací modul s mikrokontrolerem dvěma vodiči, zem GND a Tx pin na vysílací straně a Rx pin na přijímací straně. Při obousměrné komunikaci jsou využity tři vodiče, GND, Rx a Tx zapojené do kříže, toto zapojení je využito pro komunikaci mezi Car4 a ovládáním. Jedním směrem se odesílají řídicí data, ty slouží pro řízení vozidla. Druhým směrem se odesílají data ze senzorů pro informování uživatele na displeji. Další piny u linky RS232 slouží k hardwarovému řízení přenosu dat a nemusí být zapojeny. Maximální možná rychlost komunikace po sériové lince je 115200 kb/s. [1,12,13]

### 3.2 Paketace



Obr.3.1: Posílání dat mezi ovládáním a vozidlem Car4

Data, která jsou posílána po UARTu, mohou nabývat hodnot 8 bitů = 1 Byte (0-255). Mnohdy je potřeba odesílat vyšší hodnoty než osmibitové, proto byla vytvořena paketace dat pro odesílání šestnáctibitových hodnot. Do bloku „paketace“ vstupují šestnáctibitové hodnoty. Zde se rozdělí jedna šestnáctibitová hodnota na dvě osmibitové hodnoty. Nejdříve se odešle spodních osm bitů a poté se odešle horních osm bitů. K odesílaným hodnotám se přidá Start Byte, který udává začátek posílaných dat. Dále k odesílaným hodnotám přidají CRC Byty (Cyklický redundantní součet), které provádí kontrolní součet pro zjištění chyb během přenosu. Po provedení paketace se data odesílají z Tx pinu užitím bezdrátového UARTu z vozidla Car4.

V dálkovém ovládání je potřeba tyto odesílané hodnoty z vozidla Car4 přijmout a složit z nich opět šestnáctibitové čísla. Data jsou přijaty Rx pinem a v bloku „depaketace“ jsou dvě k sobě patřící osmibitové hodnoty převedeny na šestnáctibitovou hodnotu. Přijatá datová zpráva je zkontrolována, zda obsahuje Start Byte a zda se shoduje cyklický redundantní součet s přijatými CRC Byty. Pokud je přijat Start Byte a CRC je shodný, uloží se nová přijatá hodnota do paměti pomocí bloku „Data Store Write“.

Pro další práci s přijatou hodnotou je použit blok „Data Store Read“, pomocí kterého se načte přijatá uložená hodnota a lze ji dále zpracovávat.

## 4. Úprava programu dálkového ovládání

---

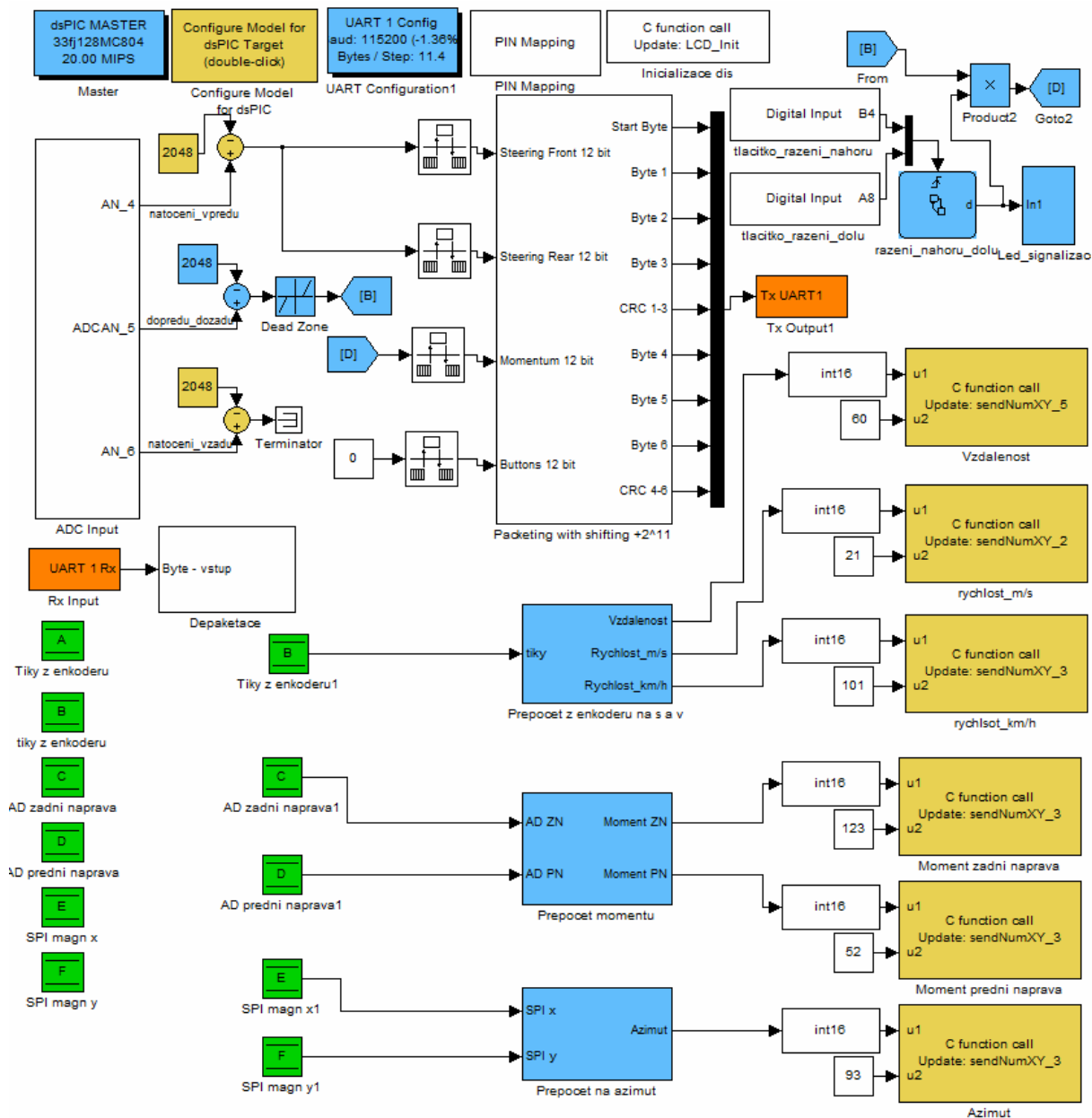
### 4.1 Současný stav – jednosměrná komunikace

Cílem práce je vytvoření programu pro dálkové ovládání, přidání obousměrné komunikace a zobrazení dat na displeji. Komunikace z ovládání do Car4 již byla vytvořena a slouží pro dálkové ovládání vozidla Car4, které se řídí pomocí změny rychlosti/momentu a ovládání přední a zadní nápravy. K tomu slouží dvojice joysticků, které jsou zapojené jako odporový dělič. Při pohybu joysticku se mění odpor a tím i napětí na potenciometru. Pomocí AD převodníku je napětí z joysticku převedeno na digitální signál, který je zpracován v mikrokontroleru. Dále jsou na ovládání tlačítka, jedna dvojice, která je v každém joysticku, slouží pro nastavení rozsahu rychlosti. Výstup z AD převodníku je vynásoben konstantou podle zvoleného rychlostního stupně. Druhá dvojice tlačítek umožňuje přepínání modu řízení, je možné řízení přední nápravy (zadní je nastavena na nulové poloze), ovládání přední a zadní nápravy jedním joystickem (natočení stejným směrem – jízda do boku, natočení zadní nápravy opačným směrem než přední), ovládání každé nápravy jiným joystickem. Pro řazení a změny modu řízení je využito Stateflow, což je doplněk Simulinku. Dále je na ovládání umístěna skupina diod, ty slouží pro informování o zvoleném zařazeném rychlostním stupni. [4]

Cílem je takto vytvořené dálkové ovládání upravit o přidání displeje, na kterém bude uživatel informován o stavech z vozidla, např. jakou jede rychlostí, ujetá vzdálenost, jakým jede směrem a jiné informace popsané v kapitole zobrazitelné údaje. Tyto informativní údaje jsou odesílány z vozidla Car4 do dálkového ovládání, ke kterému je připojen displej. Musí se vytvořit obousměrná komunikace.

Programování je možné pomocí C kódu, které je velice efektivní, ale doba tvorby programu je dlouhá a vyžaduje znalosti jazyka C. Použijeme rychlejší cestu a to využití nadstavbové části Simulinku a to Kerhuel Toolbox, který slouží k propojení Simulinku s cílovým mikrokontrolerem dsPIC.

## 4.2 Program dálkového ovládání



Program dálkové ovládání je vytvořený v Simulinku, ze kterého je poté pomocí nástroje Real-Time Workshop vygenerován soubor v C kódu. Ten se následně kompiluje kompilátorem pro mikrokontrolery dsPIC do strojového kódu a ten je pak nahrán do mikrokontroleru. Nutnou součástí je Kerhuel Toolbox, který obsahuje bloky pro použitý mikrokontroler a slouží pro propojení Simulinku a mikrokontroleru. V horní části programu je umístěno řízení vozidla, ve spodní části je pak příjem a zpracování hodnot přijatých z vozidla a jejich následný výpis na displej.

## 4.2.1 Použité konfigurační bloky

### Master

Při vytváření modelu v Simulinku je téměř nezbytné začít blokem „Master“, který je nutnou a nepostradatelnou součástí vytvořeného modelu. V tomto bloku se zvolí používaný mikrokontroler, v našem případě používáme dsPIC33FJ128MC804. Dále se zde nastavuje časový krok, frekvence externího krystalu, nastavení PLL, kde zadáváme požadovaný počet instrukcí za sekundu, v našem případě je to 20e6 MIPS<sup>1</sup>.

### UART Configuration

Dále model obsahuje komunikační bloky, kde se nastavují konfigurace použitých komunikací (SPI, I2C, CAN, UART). V našem návrhu použijeme „UART Configuration“ ve kterém se nastaví používaný UART<sup>2</sup>, který používáme pro komunikaci mezi ovládáním a vozidlem Car4. V konfiguračním bloku vybereme modul UART 1 a nastavíme přenosovou rychlost na 115200 kb/s. Blok poté vypočítá ze zadaných parametrů počet Bytů, který je schopný přenést během nejrychlejší smyčky programu.

### PIN Mapping

V bloku „PIN Mapping“ se nastavují remapovatelné piny, což jsou piny, které pevně nadefinují, na kterém pinu se má daná periferie provozovat. Přiřazují se zde jednotlivé periferie mikrokontroleru k pinům, např. Timers, PWM, QEI, UART, SPI, ECAN,... . My blok „PIN Mapping“ použijeme pro nastavení sériové komunikace UART sloužící pro komunikaci mezi dálkovým ovládáním a vozidlem Car4. Nastavíme zde vysílací Tx pin a přijímací Rx pin, podle schématu elektrického zapojení.

### C function call

Blok „C function call“ slouží k volání funkce napsané v C kódu. Jelikož některé potřebné funkce nejsou v Kerhuel Toolboxu obsažené, např. funkce pro obsluhu displeje, využije se blok „C function call“, který zavolá funkci napsané v C kódu. Soubor napsaný v C kódu by měl být umístěn ve stejném adresáři jako Simulinkovský soubor mdl, ve kterém C kód voláme. Pokud je v jiném adresáři, musíme zadat v „Simulation -> Configuration Parameters -> Real-Time Workshop -> Custom Code“ adresu, kde je soubor uložen. Název souboru s funkcemi se zde zadává vždy, i když je ve stejném adresáři jako model. Do bloku je potřeba zadat název volané funkce. V našem případě je to funkce pro inicializaci displeje „LCD\_Init“. Dále se v bloku nastavují vstupy a výstupy, které v případě inicializace displeje nejsou potřeba.

---

<sup>1</sup> MIPS = Mega Instructions Per Second

<sup>2</sup> mikrokontroler dsPIC33FJ128MC804 obsahuje dvě jednotky UART.

#### 4.2.2 Problém práce displeje s UARTem

Při počáteční práci s displejem se zkoušela jeho inicializace, výpis textu a hodnot z mikrokontroleru dálkového ovládání, ke kterému je displej připojen. Například bylo z polohy joysticku získáno napětí, které bylo převedeno pomocí 12ti bitového AD převodníku na hodnotu od 0 do 4095 a ta byla zobrazena na displeji. Při úvodní práci se nejevily žádné problémy, displej zobrazoval požadované hodnoty.

Úvodní testy s komunikací přes linku UART, pomocí které komunikuje vozidlo Car 4 a dálkové ovládání, také proběhly úspěšně. Data byla odesílána z ovládání do vozidla, opačným směrem a i oběma směry současně se data odesílala. Problém nastal při pokusu, kdy jsme chtěli zobrazovat na displej data z vozidla Car4, tedy využít komunikaci přes UART a zároveň použít displej, který se vůbec neinicializoval. Postupným odstraňováním bloků bylo zjišťováno, jestli je chyba v nějakém použitém bloku. Za „problémový“ byl označen blok „PIN Mapping“, který nám svou funkcí přemapuje nastavené piny pro displej napsané v C kódu, ten je volán při inicializaci displeje. To bylo způsobeno tím, že blok „PIN Mapping“ přiřazoval piny jednotlivým perifériím po vykonání inicializace displeje. Bylo potřeba zajistit, aby funkce pro inicializaci displeje byla vykonána po nastavení pinů v bloku „PIN Mapping“. To bylo zajištěno nastavením priority bloku, neboli pořadím v jakém bloky „pracovaly“. Nastavení priority se provede kliknutím na blok pravým tlačítkem myši a vybráno „Block Properties“ a zde nastavena Priorita bloku. Konfigurační bloky a Master dostaly nejvyšší prioritu (vykonány nejdříve), poté inicializace displeje dostala nižší prioritu. Tím bylo zajištěno, že displej byl inicializován po nastavení remapovatelných pinů v bloku „PIN Mapping“. Vše ostatní dostalo ještě nižší prioritu a problém byl vyřešen. Displej i UART pracovali společně a bylo možné zobrazovat data přijatá přes UART z vozidla Car4.

#### 4.2.3 Zpracování přijatých hodnot

Hodnoty přijaté do ovládání z vozidla Car4 Rx pinem a po depaketaci se musí zpracovat před vypsáním na displej.

Enkodér

Jako první hodnota je přijata hodnota z enkodéru, která je v ticích. Hodnota je 16ti bitová, může nabývat hodnot 0 – 65535 (datový typ uint16), jelikož kolo, na kterém je umístěn enkodér, se může otáčet i na druhou stranu, byl rozsah změněn na znaménkový a to od -32768 do 32767 (datový typ int16). Jedno otočení kola odpovídá 1060 tiků. Pokud tedy touto hodnotou podělíme přijatou hodnotu, získáme údaj počtu otáček kola. Dalším vhodným přepočtem získáme ujetou vzdálenost. K tomu použijeme vzorec pro obvod kola, kde „o“ je obvod kola, „r“ je poloměr kola a „d“ je průměr kola.

$$o = 2 \pi r = \pi d \quad (1)$$

V našem případě to ještě podělíme počtem tiků na otáčku, tedy 1060, abychom získaly přepočet na jednu otáčku. Přijatou hodnotu tedy vynásobíme pomocí bloku „Gain“, do kterého napíšeme „ $\pi d / 1060$ “ a tím získáme ujetou vzdálenost, kterou poté vypíšeme na displej.

Průměr kola byl zjištěn více způsoby. Byl změřen posuvným měřítkem. Změřením posuvným měřítkem jsme získali údaj průměr kola 0,151 m. Dále byla hodnota zjištěna

experimentálně. Vozidlem Car 4 jsme ujetí určitý počet otáček a metrem jsme změřili ujetou vzdálenost. Poté upravením a dosazením do vzorce (1) jsme spočítali průměr kola. Hodnota „n“ je počet otáček a „o“ představuje ujetou vzdálenost.

$$d = \frac{o}{n \pi} \quad (2)$$

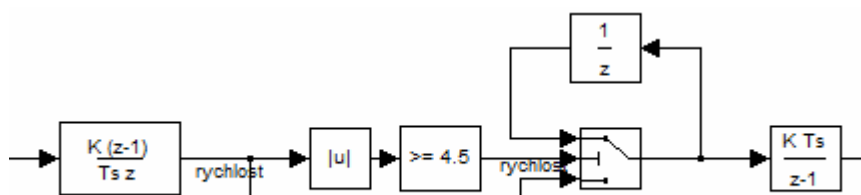
Bylo provedeno měření, při kterém se měřila vzdálenost při určitém počtu otáček, hodnoty pro různé otáčky jsou uvedeny v následující tabulce:

| Otáčky | Vzdálenost | Průměr kola |
|--------|------------|-------------|
| 1      | 0,465      | 0,148       |
| 2      | 0,938      | 0,149       |
| 3      | 1,426      | 0,151       |
| 4      | 1,855      | 0,148       |
| 5      | 2,323      | 0,148       |
| 6      | 2,72       | 0,144       |
| Průměr |            | 0,148       |

Tab 3.: Průměr kola

Po provedení aritmetického průměru jsme zvolily pro průměr kola hodnotu: 0,148 m.

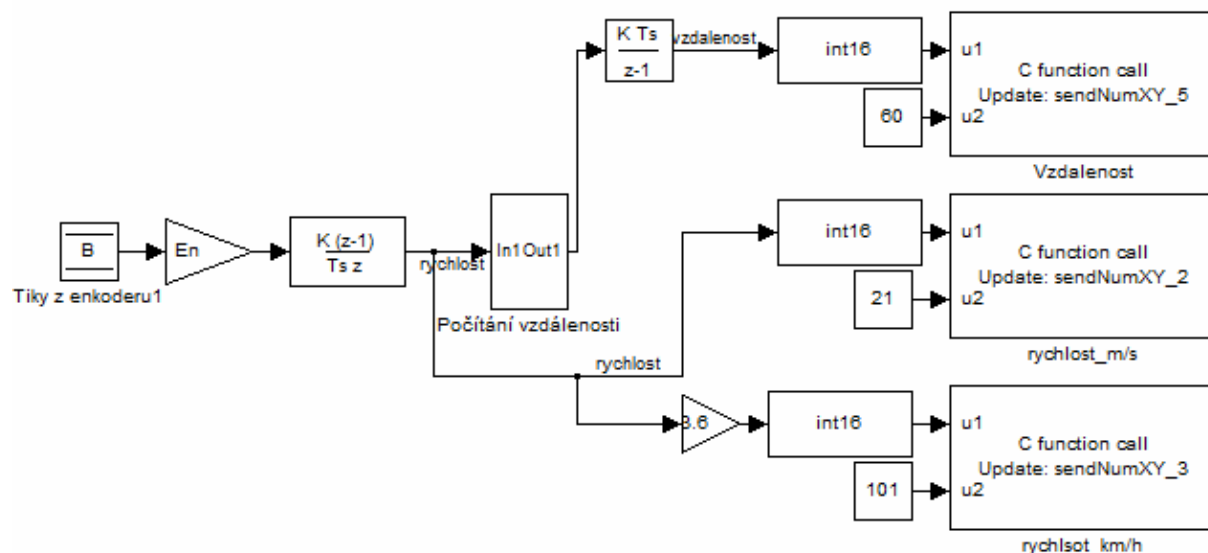
Problém nastane při získání poslední hodnoty z enkodéru, poté posílá tiky znovu od nuly. Což zapříčiní, že i ujetá vzdálenost se nám vynuluje. To by nám nevadilo, pokud by to byla dostatečně velká vzdálenost, kterou vozidlem Car4 nenajedeme. Z enkodéru přijmeme hodnotu  $2^{15}$  v jednom směru a při vynásobení přepočtem na otáčky zjistíme, že po ujetí 14 metrů se nám ujetá vzdálenost vynuluje. Tento problém byl vyřešen derivací (je získána rychlost) a následnou integrací (je získána opět vzdálenost). Při „nulování enkodéru“ je rychlost nepřiměřeně velká, aktivuje přepínač umístěný mezi derivací a integrací, který v tomto okamžiku přepne do druhé polohy. Do této polohy je přivedena hodnota o jeden časový krok zpět, a to nám zajistí přičítání vzdálenosti v době, kdy byla nulována. Po tomto kroku se přepne přepínač zpět a rychlost je integrována.



Obr.4.1: Přičítání vzdálenosti

Pro zobrazení rychlosti derivujeme ujetou vzdálenost. Zde už nevadí, když hodnota přeteče. Pro derivování nemůžeme použít klasickou numerickou derivaci, ale musíme použít derivaci diskretní. Po derivování získáme informaci o rychlosti jakou jede vozidlo Car4. Maximální rychlost, kterou vozidlo Car4 může jet, je  $4,2 \text{ ms}^{-1}$ . Vypsání ujeté vzdálenosti je v metrech na pět znaků. Rychlost je zobrazena v m/s ve formátu „X.X“ a je zobrazena i km/h ve formátu „XX.X“.





Obr.4.2: Výpočet vzdálenosti, rychlosti a vypsání na displej

### Proudový snímač LEM

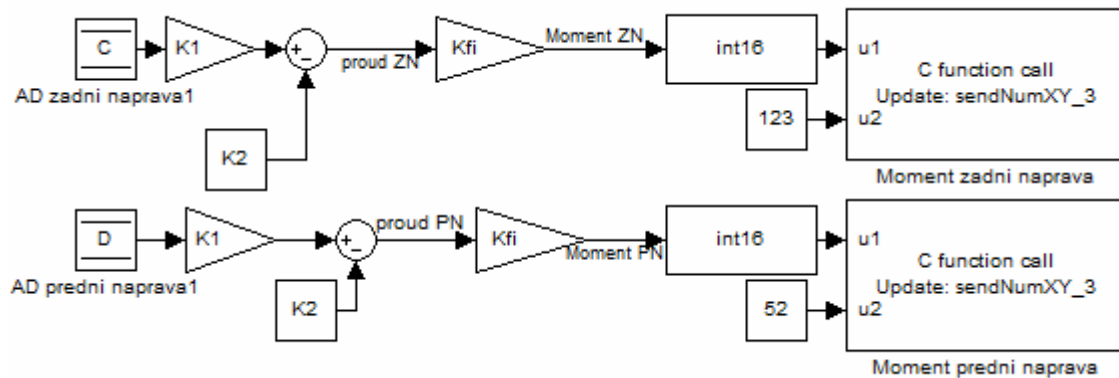
Výstupní napětí z proudového čidla je v rozmezí 0-4,5V. Mikrokontroler pracuje s napětím 3,3V. Napětí z čidla je pomocí odporového děliče přepočteno, převedeno 12ti bitovým AD převodníkem a posíláno do dálkového ovládání. Toto napětí je upraveno a přepočteno na proud procházející motorem. Přijatá hodnota může tedy nabývat hodnot 0-4095. Velikost napětí přijaté mikrokontrolerem se spočítá podle vztahu (3).

$$U = AD \frac{3,3}{4096} \quad (3)$$

AD je hodnota z AD převodníku. Hodnota 3,3 je maximální možné přijaté napětí z čidla po přepočtu odporovým děličem a hodnota 4096 je rozlišení převodníku. Od této hodnoty se odečte napětí, při kterém je nulový proud, a to pro získání nulové hodnoty. Z voltampérové charakteristiky čidla je tato hodnota 2,5 V, ta je vynásobena odporovým děličem. Pro získání proudu je celý výpočet podělen konstantou udávající závislost proudu na napětí u proudového čidla LEM. Rovnice pro proud je uvedena ve vztahu (4).

$$I = \frac{AD \frac{3,3}{4096} - 2,5 \frac{3,3}{4,5}}{0,052083} \quad (4)$$

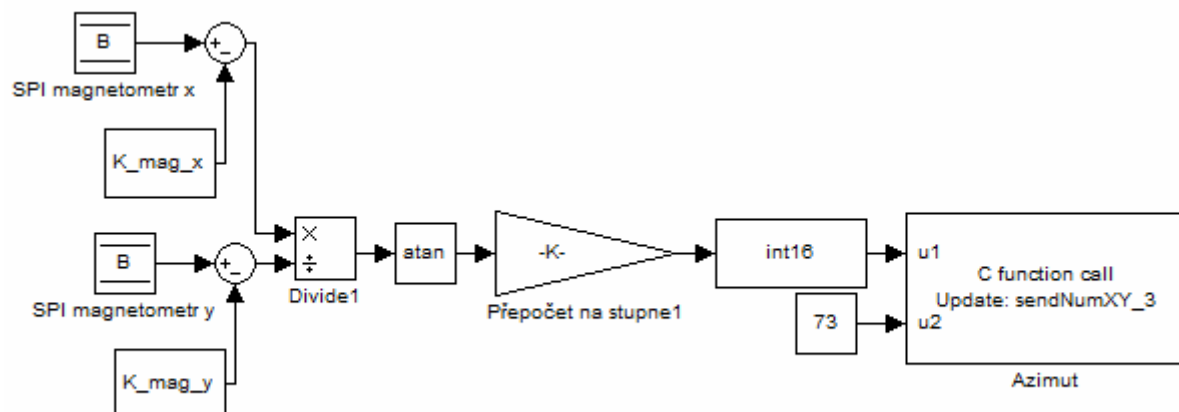
Po vynásobení konstantou motoru (napěťová a magnetický tok) získáme záběrný moment motoru. Ten je zobrazen na displeji u přední a u zadní nápravy. Moment může dosáhnout maximální hodnoty 3,027 Nm. Zvolili jsme proto formát zápisu na dvě desetinná místa. (X.XX)



Obr.4.3: Výpočet proudu, momentu a vypsání na displej

## ADIS 16045

Z inerciální jednotky ADIS je z magnetometru vypočten azimut udávající jakým směrem jede vozidlo Car4. Při  $0^\circ$  jede na sever. Z osy „x“ a „y“ magnetometru jsou přijaty hodnoty. Užitím kompasu jsou zkalibrovány. Dále pomocí goniometrické funkce tangens je vypočten azimut, který je v radiánech. Po vynásobení „ $180/\pi$ “ je azimut ve stupních.

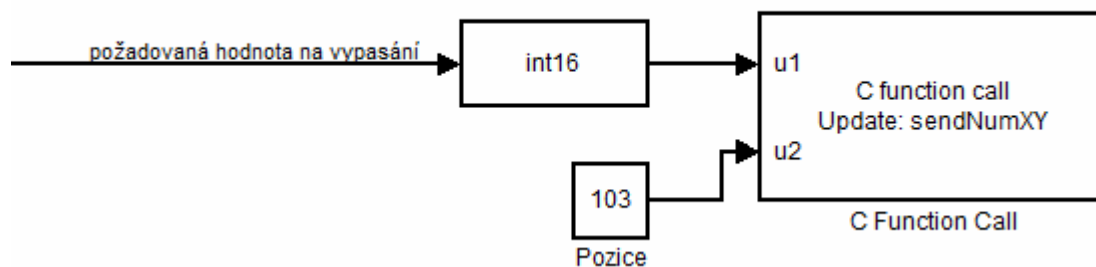


Obr.4.4: Výpočet azimutu a vypsání na displej

### 4.2.4 Vypisování hodnot na displej

Přijaté a zpracované hodnoty potřebujeme vypsát na displej dálkového ovládání. Jak už bylo zmíněno Kerhuel Toolbox neobsahuje funkce pro práci s displejem. Využijeme znovu blok „C function call“, který nám zavolá funkci napsanou v C kódu. Zavolaná funkce nám zajistí vypsání požadované hodnoty na displej. Název funkce pro vypsání požadované hodnoty je „sendNumXY“. Blok pro vypsání hodnoty na displej, který volá funkci napsanou v C kódu, obsahuje dva vstupy, které musíme nastavit v parametrech bloku. Do vstupu jedna přivádíme hodnotu, kterou chceme vypsát na displej (hodnota je datového typu int16). A do druhého vstupu přivádíme hodnotu, ta udává, na kterou pozici se má hodnota na vstupu jedna vypsát (datový typ int8). Hodnotu udávající pozici zadáváme ve tvaru sloupec řádek. Např. bude-li hodnota „103“, začne displej vypisovat na jedenáctý sloupec a čtvrtý řádek. Funkce v C kódu pro inicializaci a vypsání údajů na displej byly převzaty z bakalářské práce [3] a byly

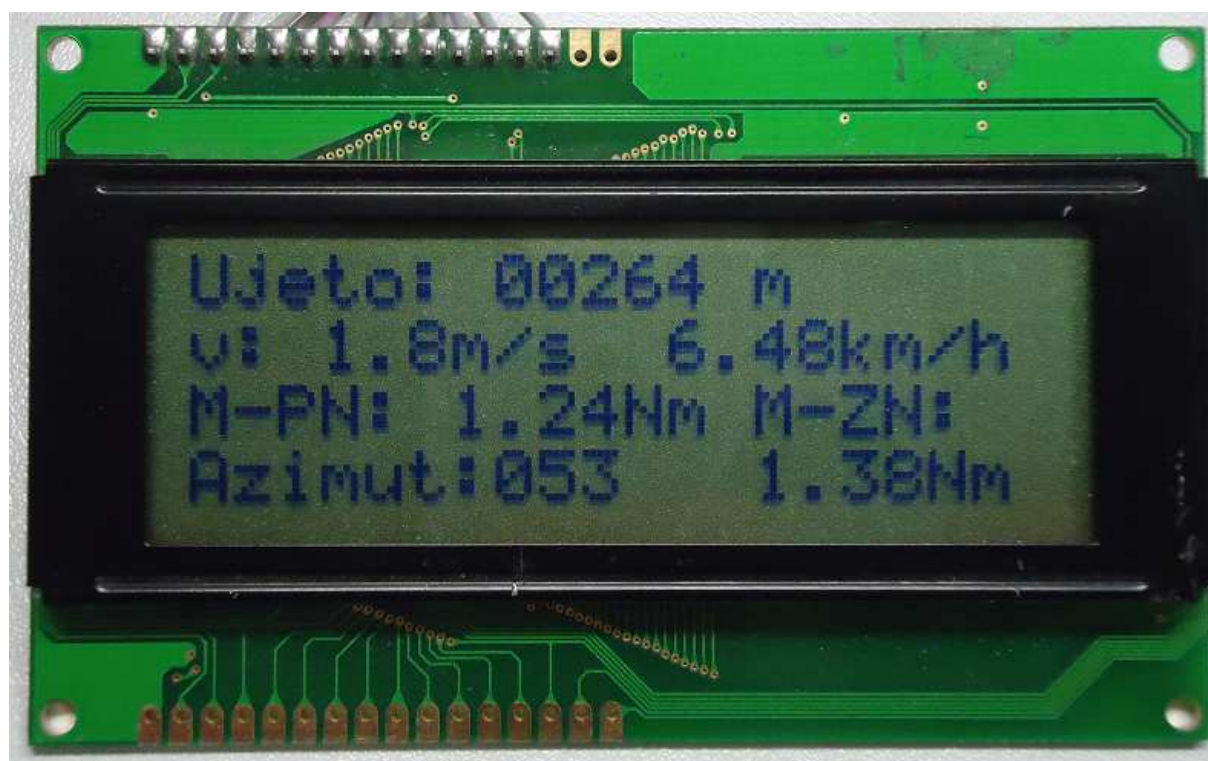
upraveny pro použití s naším mikrokontrolerem. Jelikož naše vypisované údaje mají rozdílnou délku a počet desetinných míst. Je pro každou vypisovaný údaj volána „jiná“ funkce pro zajištění potřebného počtu znaků a správné umístění desetinné čárky.



Obr. 4.5: Vypsání hodnot na displej



Obr.4.6: Dálkové ovládání s displejem



Obr.4.7: Displej s vypsanými hodnotami

## 5. Úprava palubní elektroniky

### 5.1 Akumulátory pro napájení

Dalším cílem této práce je úprava kabeláže, která slouží k napájení vozidla Car4 a pro nabíjení akumulátorů. Experimentální vozidlo Car4 obsahuje čtveřici akumulátorů.

K napájení stejnosměrných motorů (na vozidle Car4 jsou umístěny čtyři DC motory s enkodéry ovládané H-můstky) je využita dvojice lithium-železných akumulátorů A123, 13,2 V, kapacita 2300 mAh, 4 články. Dvojice akumulátorů pro napájení DC motorů je zapojena do série pro získání potřebného napětí.

K napájení elektroniky jsou používány nikl-metal hydridové akumulátory, 6V, kapacita 2700 mAh. K napájení serv je použit stejný akumulátor jako pro napájení elektroniky.

### 5.2 Původní napájení

Před úpravou kabeláže se spouštělo napájení elektroniky a serv spojením konektorů mezi akumulátorem a elektronikou, popřípadě servy. Při nabíjení akumulátoru se opět spojily konektory, tentokrát akumulátoru s nabíječkou.

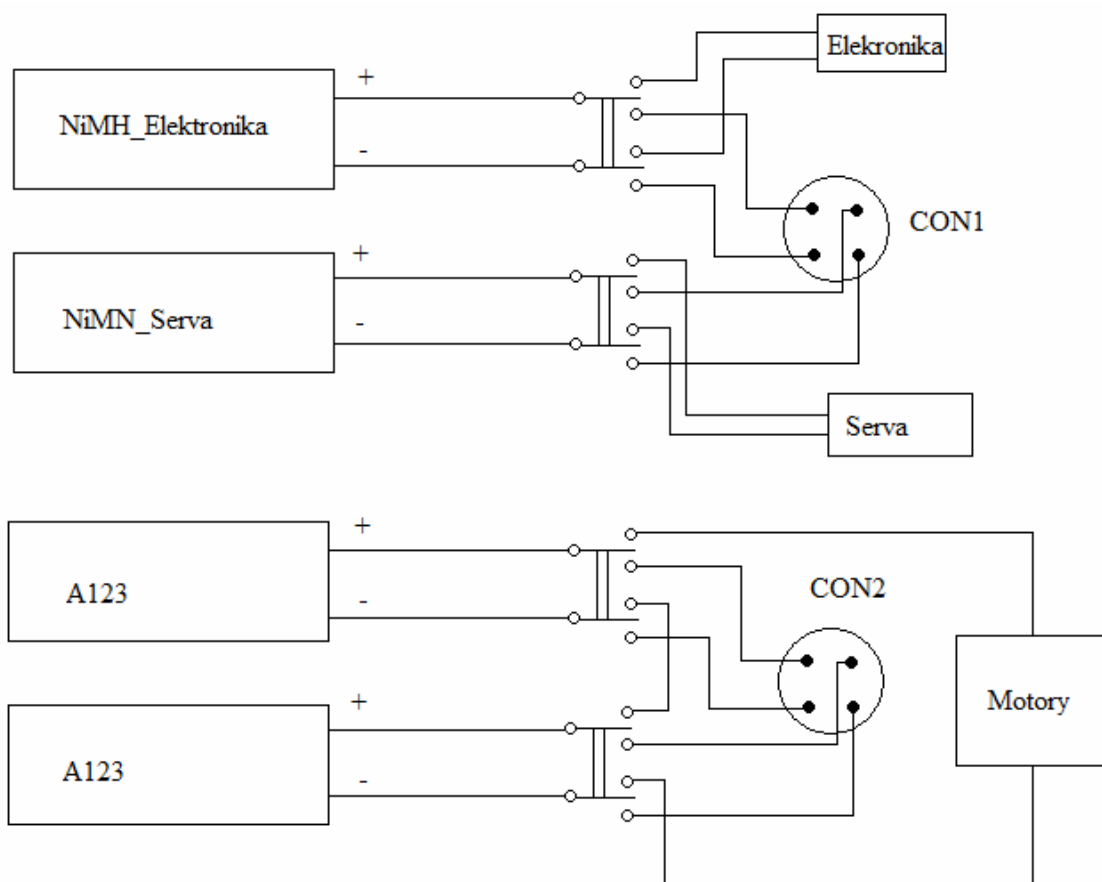
DC motory byly spínány pomocí spínače, ale při potřebě nabít akumulátory se musely v podvozku vozidla rozpojit sériově zapojené akumulátory a každý zvlášť nabít. Po nabití znovu připojit a zaizolovat. Což při častém nabíjení je nepraktický a zdlouhavý proces.

### 5.3 Napájení pomocí spínačů

Základem úpravy kabeláže je vytvořit jednoduché přepínání, které umožní stavy sepnuto – vypnuto – nabíjení. K tomuto účelu byly využity čtyři kolébkové spínače, které mají polohy ON – OFF – ON. Každý akumulátor má svůj spínač a jednoduchým způsobem se sepnou napájení elektroniky, serv a motorů. U motorů, aby bylo možné nabíjení akumulátoru přepnutím spínače, musí být ve stavu „sepnuto“ sepnuty oba spínače, jelikož jsou motory zapojeny do série. Při sepnutí jednoho spínače, by nebyl sepnut obvod a motory by nebyly napájeny.

Pro nabíjení je spínač přepnut do druhé polohy ON do stavu „nabíjení“. Akumulátor je připojen ke konektoru CON. K němu je připojena externí nabíječka a může začít proces nabíjení. Konektor CON je použit čtyř pinový a akumulátor je připojen pomocí dvou pinů (plus a minus). Na jeden konektor CON jsou připojeny dva akumulátory. Pomocí přepínače na kabeláži, která slouží pro připojení nabíječky ke konektoru CON, přepínáme nabíjení jedné nebo druhé baterie. Na první konektor jsou připojeny akumulátory pro elektroniku a pro serva. Na druhý konektor jsou připojeny dva akumulátory pro DC motory.

Nabíjení akumulátorů je řešeno pomocí externí nabíječky Hyperion EOS0610i. Akumulátory DC motorů byly umístěny podvozku vozidla, do podvozku jsme přidali i akumulátory pro serva a elektroniku. V zadní části vozidla Car4 jsme umístili „krabičku“, ve které jsou zabudované spínače a nabíjecí konektory.



Obr.5.1: Schéma napájení



Obr.5.2: Krabíčka se spínači

# Závěr

---

Hlavním cílem této práce bylo připojení displeje k dálkovému ovládání a zobrazení hodnot. Elektroniku pro připojení displeje nebylo potřeba nijak upravovat. Pro připojení byl využit desetipinový digitální výstup na mikrokontroleru, ke kterému byl pomocí vytvořené kabeláže připojen displej.

Informace zobrazené na displeji jsou posílány z vozidla Car4. Pro umožnění posílání hodnot byla přidána obousměrná komunikace mezi dálkovým ovládáním a vozidlem Car4. Problém nastal při vypisování hodnot posílaných přes UART z vozidla Car4. Displej nebyl inicializován, přiřazení jeho pinů bylo přednastaveno piny užívající UART. To bylo vyřešeno nastavením priority bloků. Inicializace byla nastavena po vykonání natavení pinů pro UART.

Pro vypisování na displej byly zvoleny údaje z enkodéru, z proudového čidla LEM a z inerciální jednotky ADIS, která obsahuje magnetometr, gyroskop a akcelerometr. Z enkodéru je zobrazena ujetá vzdálenost a rychlosti jízdy vozidla Car4. Z proudového čidla je vypočítaný a zobrazený záběrný moment přední a zadní nápravy. Z ADISU je využit magnetometr, ze kterého je vypočten a zobrazen azimut, neboli směr jakým vozidlo jede. Další zobrazitelné údaje jsou popsány v práci.

Dále byla provedena úprava zapojení palubní elektroniky vozidla Car4, na kterém byly umístěny čtyři třípolohové spínače. Každý akumulátor má svůj spínač umožňující přepínání mezi polohami zapnuto – vypnuto – nabíjení.



# Literatura

---

- [1] VEJLUPEK, J.: *Vývoj elektroniky pro řízení trakce experimentálního vozidla*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. Vedoucí diplomové práce Ing. Robert Grepl, Ph.D.
- [2] ŠIMURDA, M.: *Návrh a realizace doplňkového sensorického systému pro experimentální vozidlo*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. Vedoucí bakalářské práce Ing. Robert Grepl, Ph.D.
- [3] SOVA, V.: *Testovací jednotka proporcionálních ventilů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. 50 s. Vedoucí bakalářské práce Ing. Robert Grepl, Ph.D..
- [4] KLIMEŠ, D.: *Generování kódu pro mikrokontroléry pomocí automatických nástrojů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2011. Vedoucí bakalářské práce Ing. Josef Vejlupek.
- [5] GM Electronic, *Displeje* [online]. 2012-05-09  
Dostupné z <http://www.gme.cz/displeje/>
- [6] HW, *Inteligentní displeje a jejich připojení k PC* [online]. 2012-05-09  
Dostupné z: [http://www.hw.cz/docs/lcd\\_iq\\_displaye/lcd\\_iq\\_dip.html](http://www.hw.cz/docs/lcd_iq_displaye/lcd_iq_dip.html)
- [7] Robotika, *Enkodéry* [online]. 2012-05-09  
Dostupné z <http://robotika.cz/guide/encoders/en>
- [8] Wikipedia, *Hallův jev* [online]. 2012-05-09  
Dostupné z [http://cs.wikipedia.org/wiki/Hall%C5%AFv\\_jev](http://cs.wikipedia.org/wiki/Hall%C5%AFv_jev)
- [9] Data-sheet, *LTS 25-NP* [online]. 2012-05-09  
Dostupné z <http://www.vo.gme.cz/dokumentace/985/985-037/dsh.985-037.1.pdf>
- [10] Automatizace.hw, *Lze jedním senzorem měřit zrychlení, rotaci i magnet. pole* [online]  
Dostupné z <http://automatizace.hw.cz/lze-jednim-senzorem-merit-zrychleni-rotaci-i-magnet-pole>



- [11] Omegaeng, *Úvod do snímačů zrychlení* [online]. 2012-05-09  
Dostupné z <http://www.omegaeng.cz/prodinfo/Accelerometers.html>
  
- [12] Advanced Radio Telemetry, *HW 86010* [online]. 2012-05-09  
Dostupné z <http://www.artbrno.cz/index.php/p1rm05/p1rm05c01/102-hw-86010>
  
- [13] Root, *Komunikace pomocí sériového portu RS-232C(po druhé)* [online]. 2012-05-09  
Dostupné z [http://www.root.cz/clanky/komunikace-pomoci-serioveho-portu-rs-232c\(-podruhe\)](http://www.root.cz/clanky/komunikace-pomoci-serioveho-portu-rs-232c(-podruhe))